

Today



- Finishing up inter-domain routing
 - ◆ Economics of peering/settlement
- Review of end-to-end forwarding
- How we build routers



A History of Settlement

- The telephone world
 - ◆ LECs (local exchange carriers) (e.g., pacbell, nynex)
 - ◆ IXC (inter-exchange carriers) (e.g., sprint, at&t)
- LECs MUST provide IXCs access to customers
 - ◆ This is enforced by laws and regulation
- When a call goes from one phone company to another:
 - ◆ Call billed to the caller
 - ◆ The money is split up among the phone systems – this is called “settlement”



Business Relationships

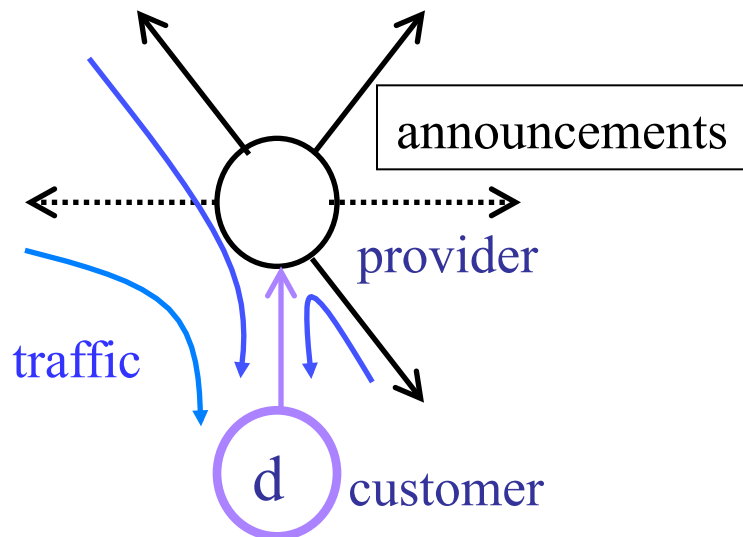
- Neighboring ASes have business contracts
 - ◆ How much traffic to carry
 - ◆ Which destinations to reach
 - ◆ How much money to pay
- Common business relationships
 - ◆ Customer-provider
 - » E.g., Princeton is a customer of USLEC
 - » E.g., MIT is a customer of Level3
 - ◆ Peer-peer
 - » E.g., UUNET is a peer of Sprint
 - » E.g., Harvard is a peer of Harvard Business School



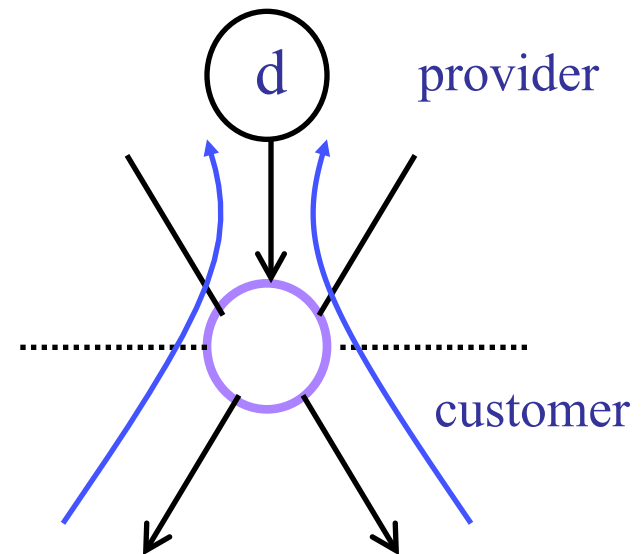
Customer/Provider

- Customer needs to be reachable from everyone
 - ◆ Provider tells all neighbors how to reach the customer
- Customer does not want to provide transit service
 - ◆ Customer does not let its providers route through it

Traffic **to** the customer



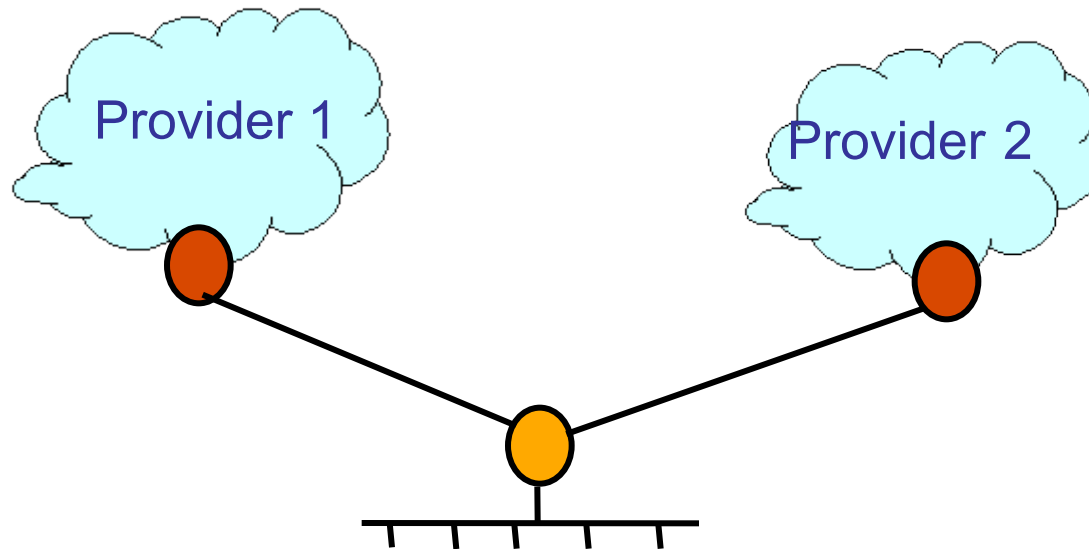
Traffic **from** the customer





Multi-Homing

- Customers may have more than one provider
 - ◆ Extra reliability, survive single ISP failure
 - ◆ Financial leverage through competition
 - ◆ Better performance by selecting better path
 - ◆ Gaming the 95th-percentile billing model

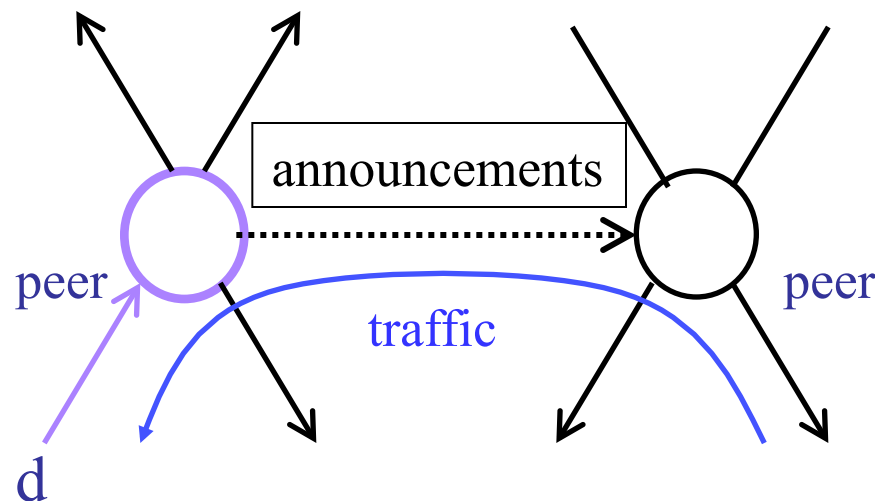




Peer-to-Peer Relationship

- Peers exchange traffic between customers
 - ◆ AS exports *only* customer routes to a peer
 - ◆ AS exports a peer's routes *only* to its customers
 - ◆ Often the relationship is settlement-free (i.e., no \$\$\$)

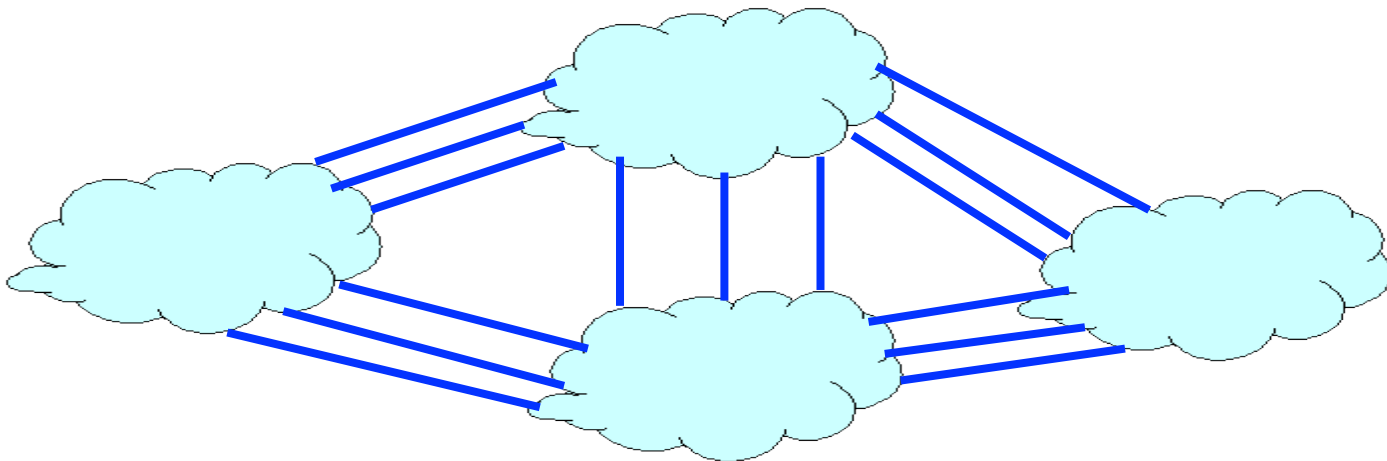
Traffic to/from the peer and its customers



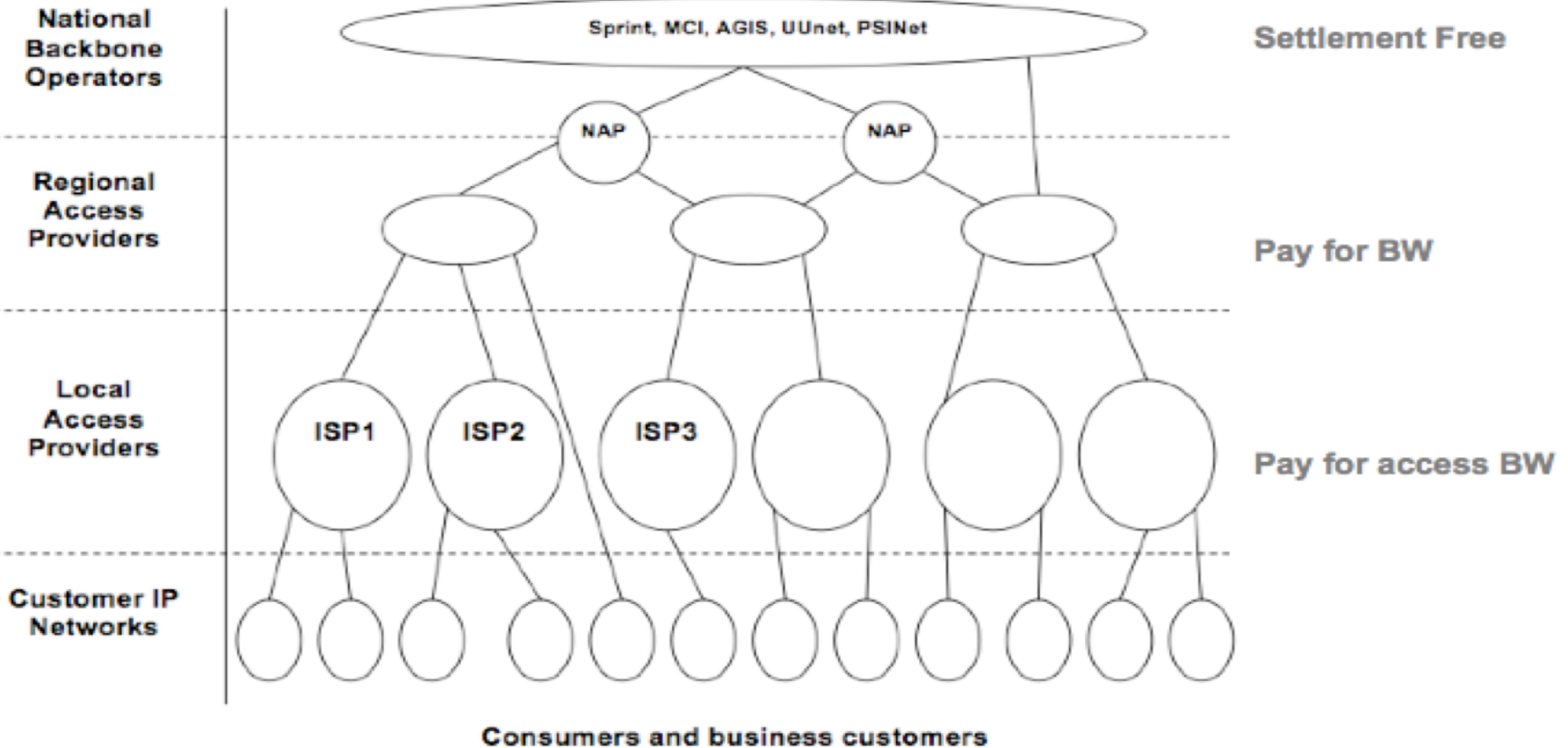


Tier-1 Providers

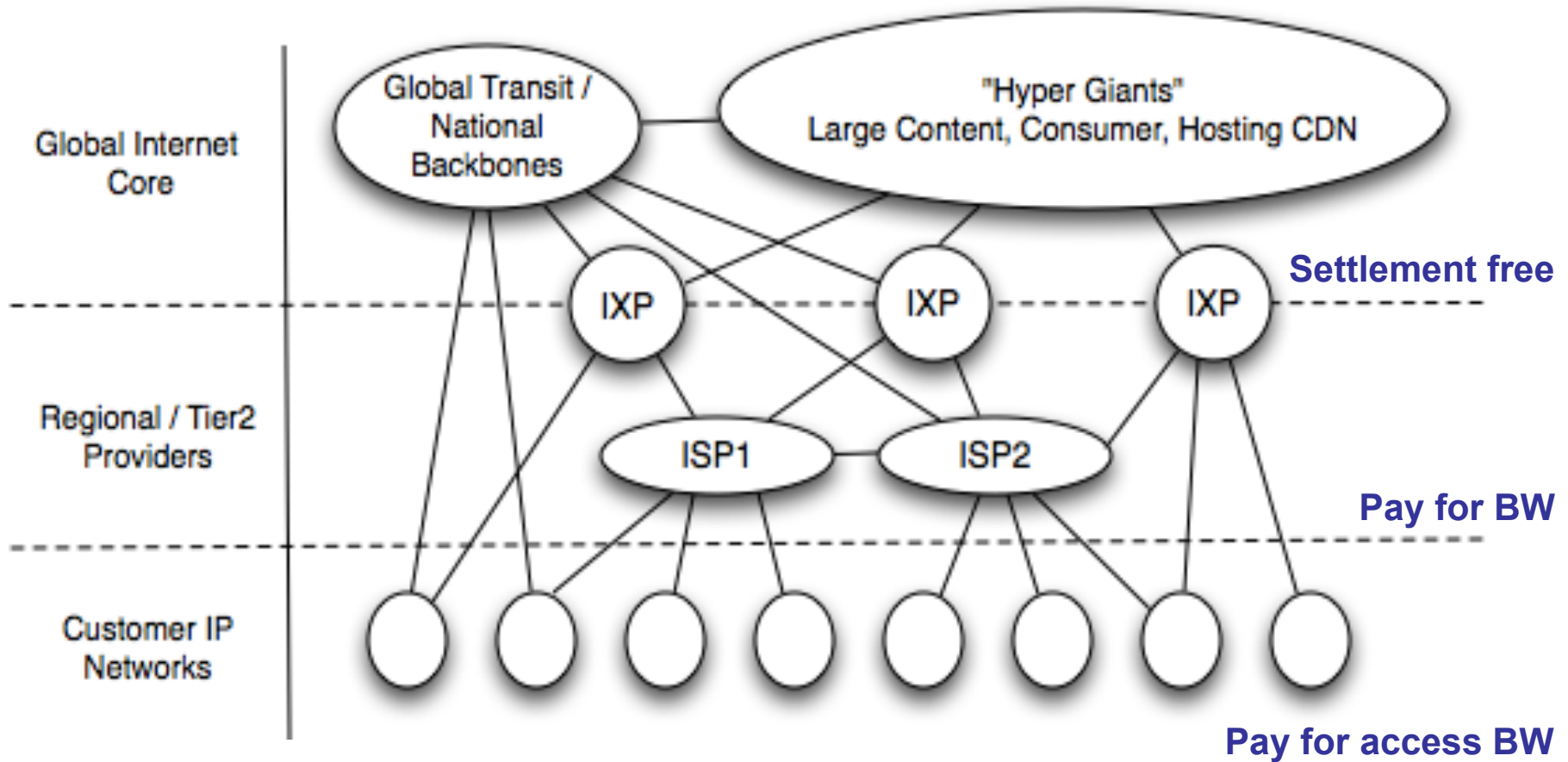
- Make up the “core” of the Internet
 - ◆ Has no upstream provider of its own
 - ◆ Typically has a national or international backbone
- Top of the Internet hierarchy of ~10-20 ASes
 - ◆ E.g., AT&T, Level3, NTT, Qwest, SAVVIS (formerly Cable & Wireless), Sprint, Verizon
 - ◆ Full peer-peer connections between tier-1 providers



Traditional “Tiered Internet”



The New Reality





BGP Summary

- Interdomain-routing
 - ◆ Exchange reachability information (plus hints)
 - ◆ BGP is based on path vector routing
 - ◆ Local policy to decide which path to follow
- Traffic exchange policies are a big issue \$\$\$
 - ◆ Complicated by lack of compelling economic model (who creates value?)
 - ◆ Can have significant impact on performance

Lecture 17:

Router Design

CSE 123: Computer Networks
Stefan Savage



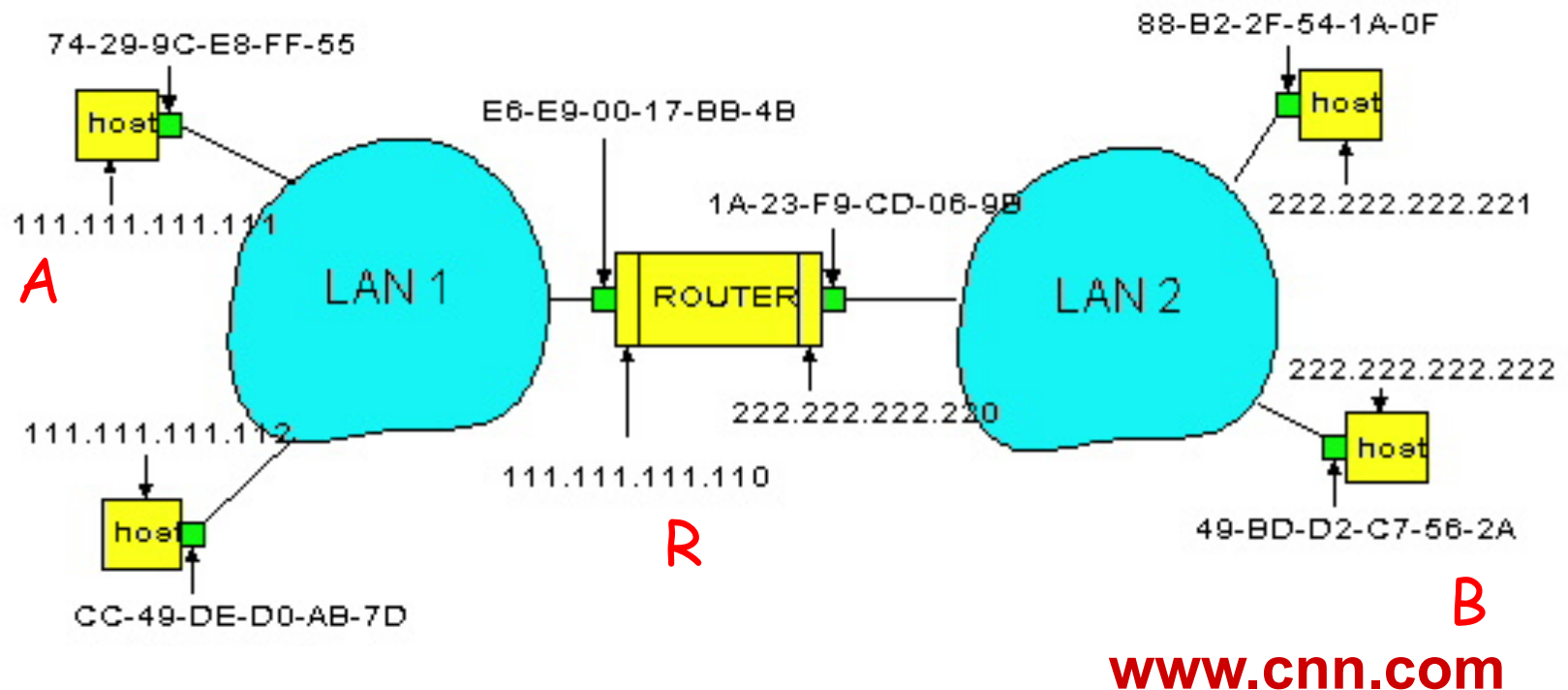


Lecture 14 Overview

- End-to-end lookup and forwarding example
- Router internals
 - ◆ Buffering
 - ◆ Scheduling



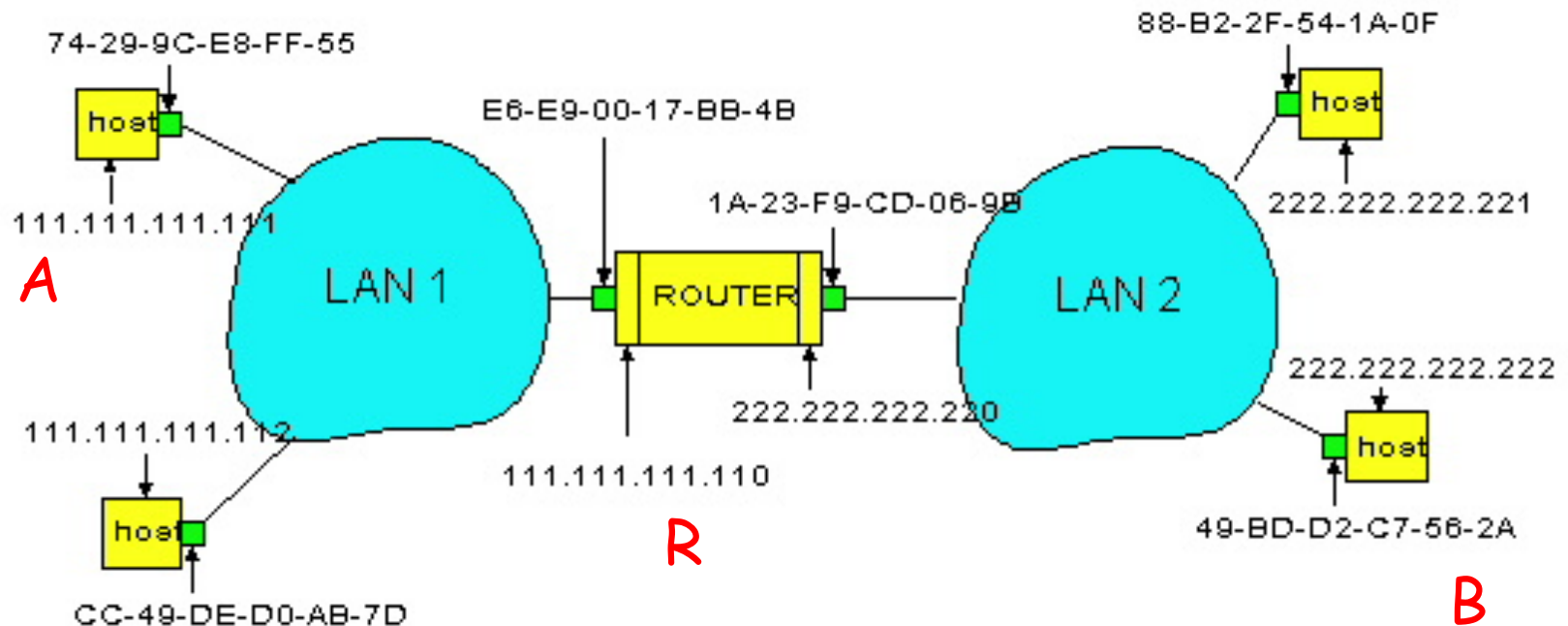
Example: Sending to CNN





Basic Steps

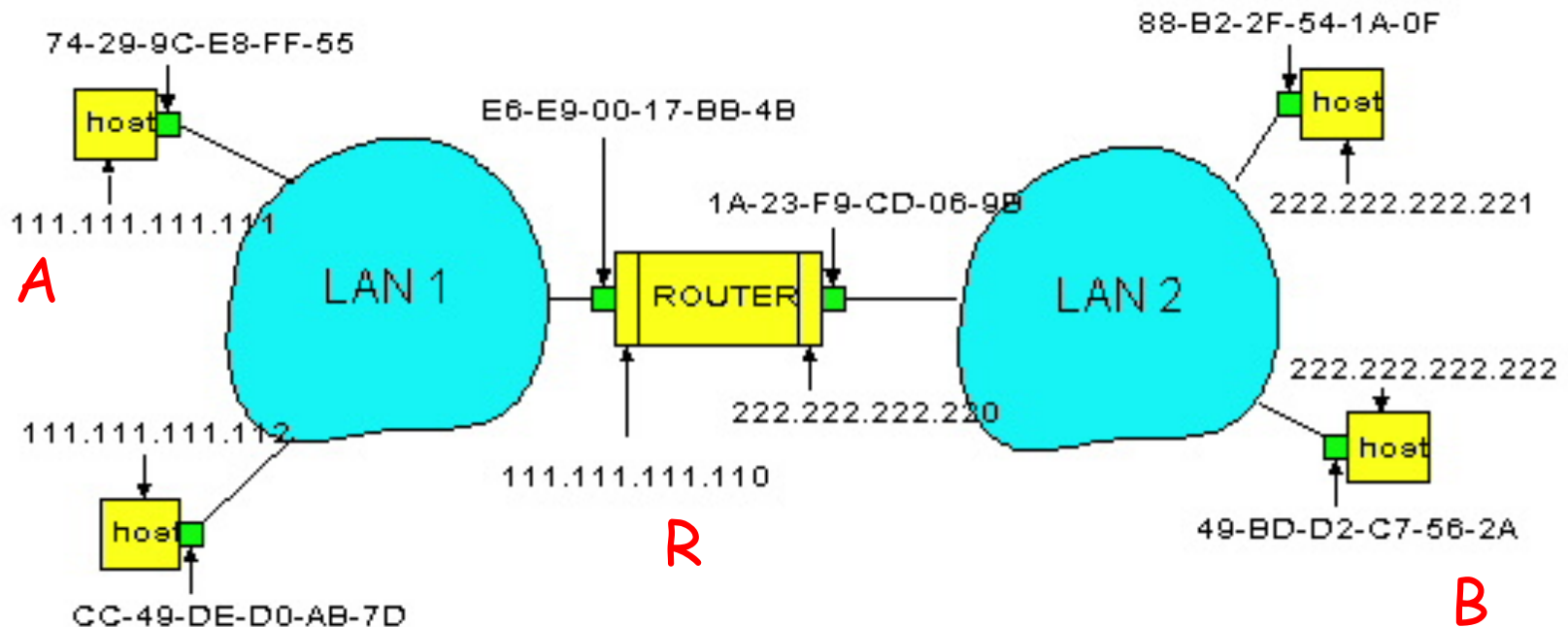
1. Host *A* must learn the IP address of *B* via DNS
2. Host *A* uses gateway *R* to reach external hosts
3. Router *R* forwards IP packet to outgoing interface
4. Router *R* learns *B*'s MAC address and forwards frame



Host A Learns B's IP Address



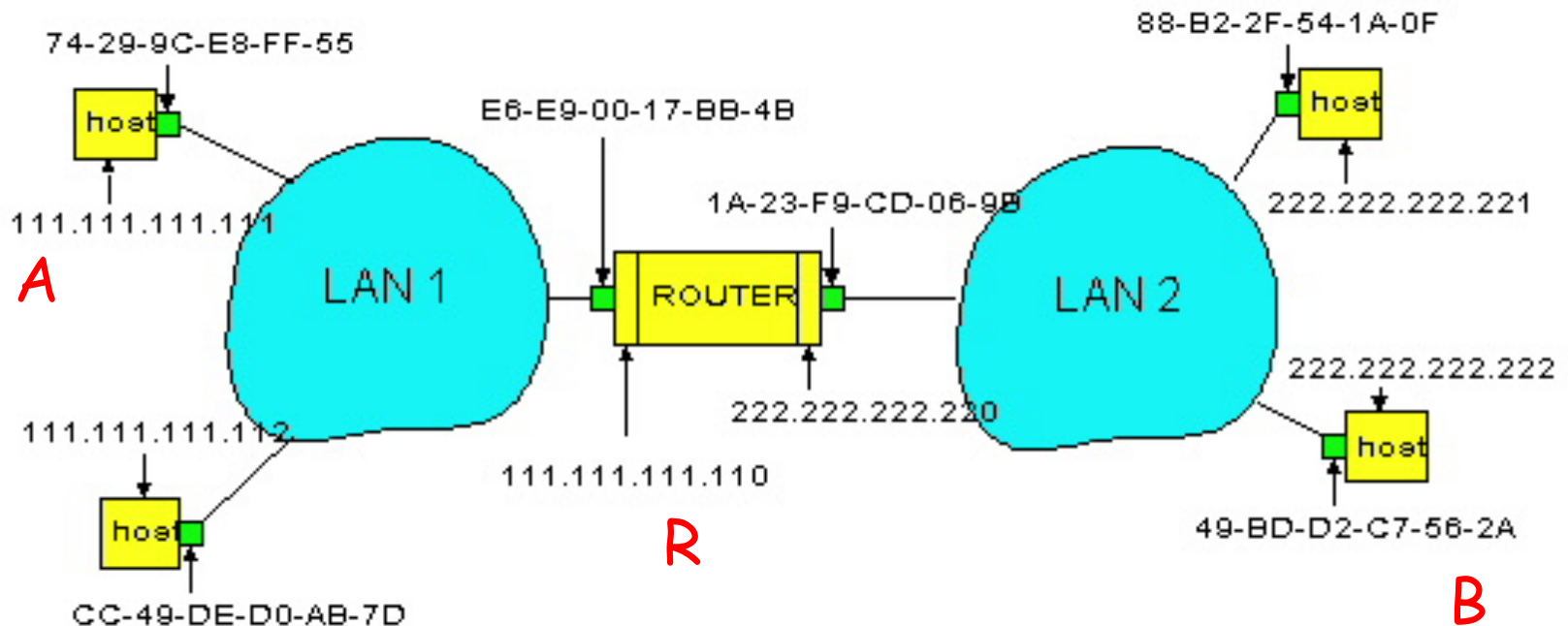
- Host A does a DNS query to learn *B*'s address
 - ◆ Suppose `gethostbyname()` returns 222.222.222.222
- Host A constructs an IP packet to send to *B*
 - ◆ Source 111.111.111.111, dest 222.222.222.222



Host A Learns B's IP Address



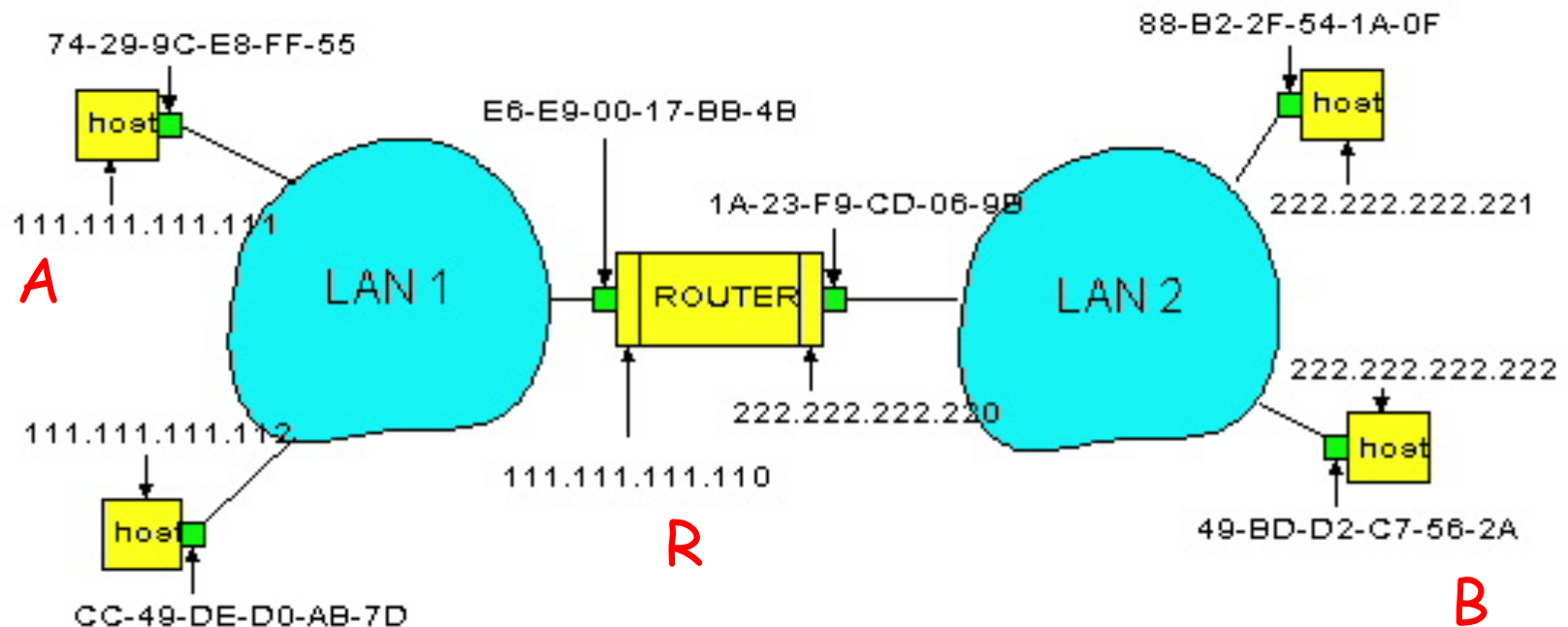
- IP packet
 - ◆ From A: 111.111.111.111
 - ◆ To B: **222.222.222.222**
- Ethernet frame
 - ◆ From A: 74-29-9C-E8-FF-55
 - ◆ To gateway: ????



A Decides to Send Through *R*



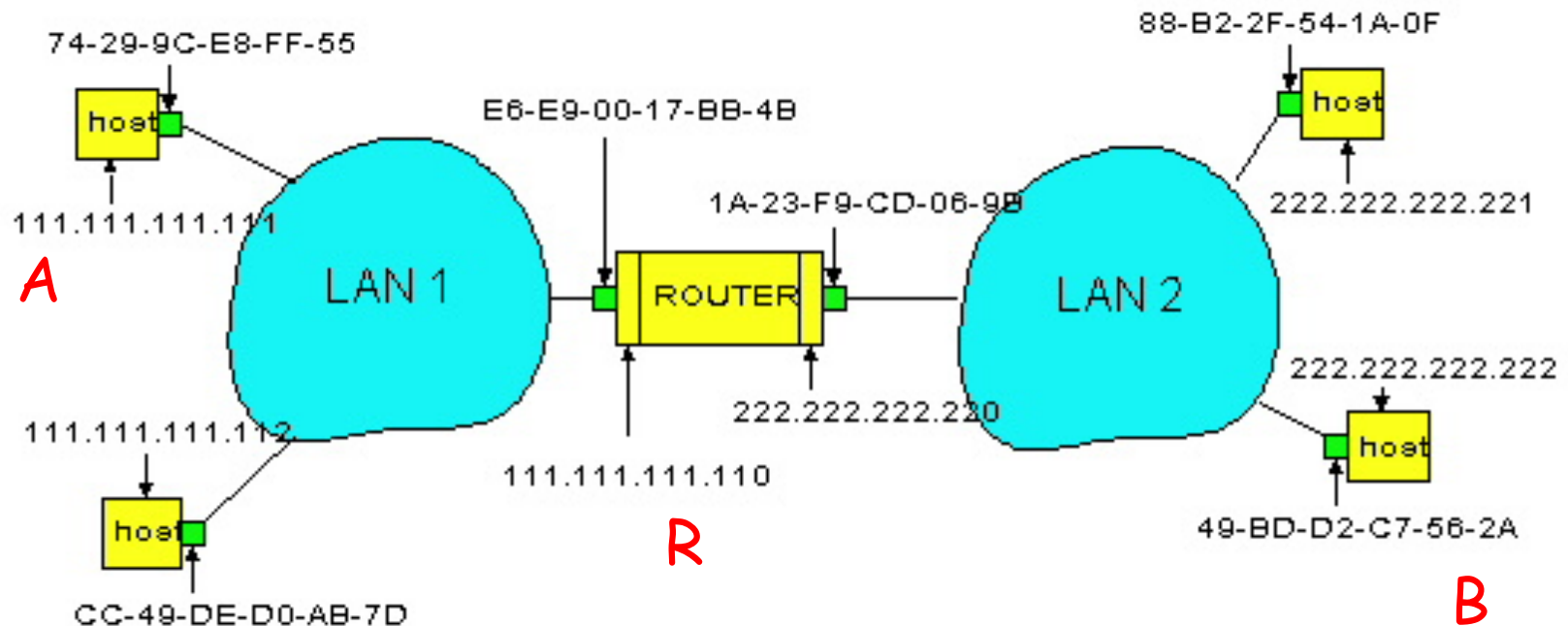
- Host *A* has a gateway router *R*
 - ◆ Used to reach destds outside of 111.111.111.0/24
 - ◆ Address 111.111.111.110 for *R* learned via DHCP
- But, what is the MAC address of the gateway?



A Sends Packet Through R



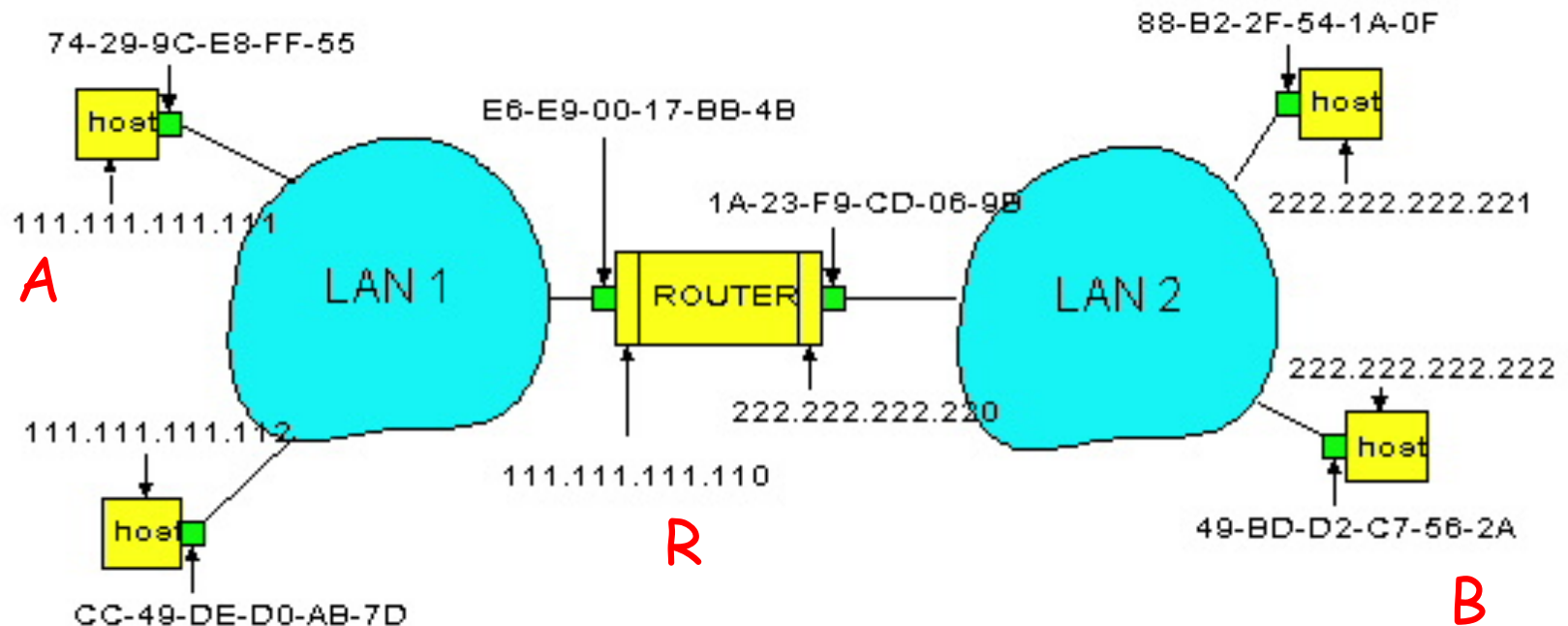
- Host A learns the MAC address of *R*'s interface
 - ◆ ARP request: broadcast request for 111.111.111.110
 - ◆ ARP response: R responds with E6-E9-00-17-BB-4B
- Host A encapsulates the packet and sends to *R*



A Sends Packet Through R



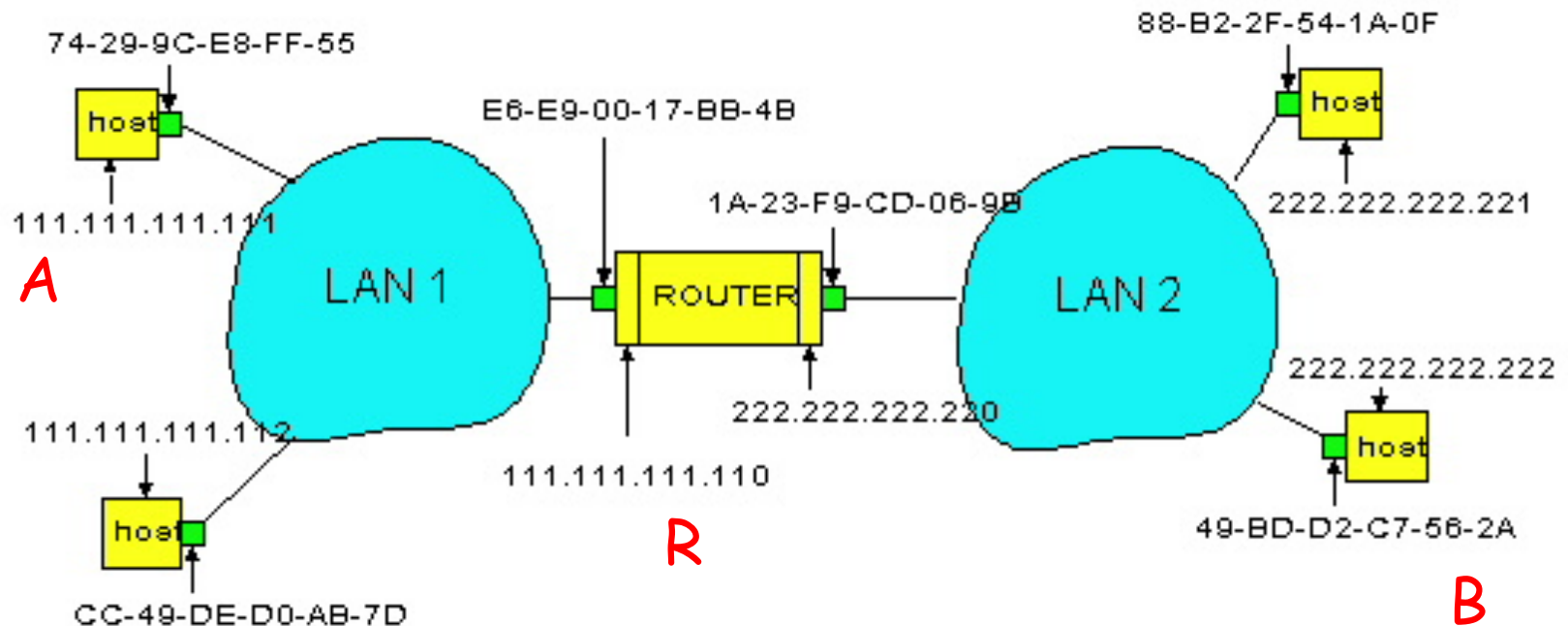
- IP packet
 - ◆ From A: 111.111.111.111
 - ◆ To B: 222.222.222.222
- Ethernet frame
 - ◆ From A: 74-29-9C-E8-FF-55
 - ◆ To R: E6-E9-00-17-BB-4B





R Looks up Next Hop

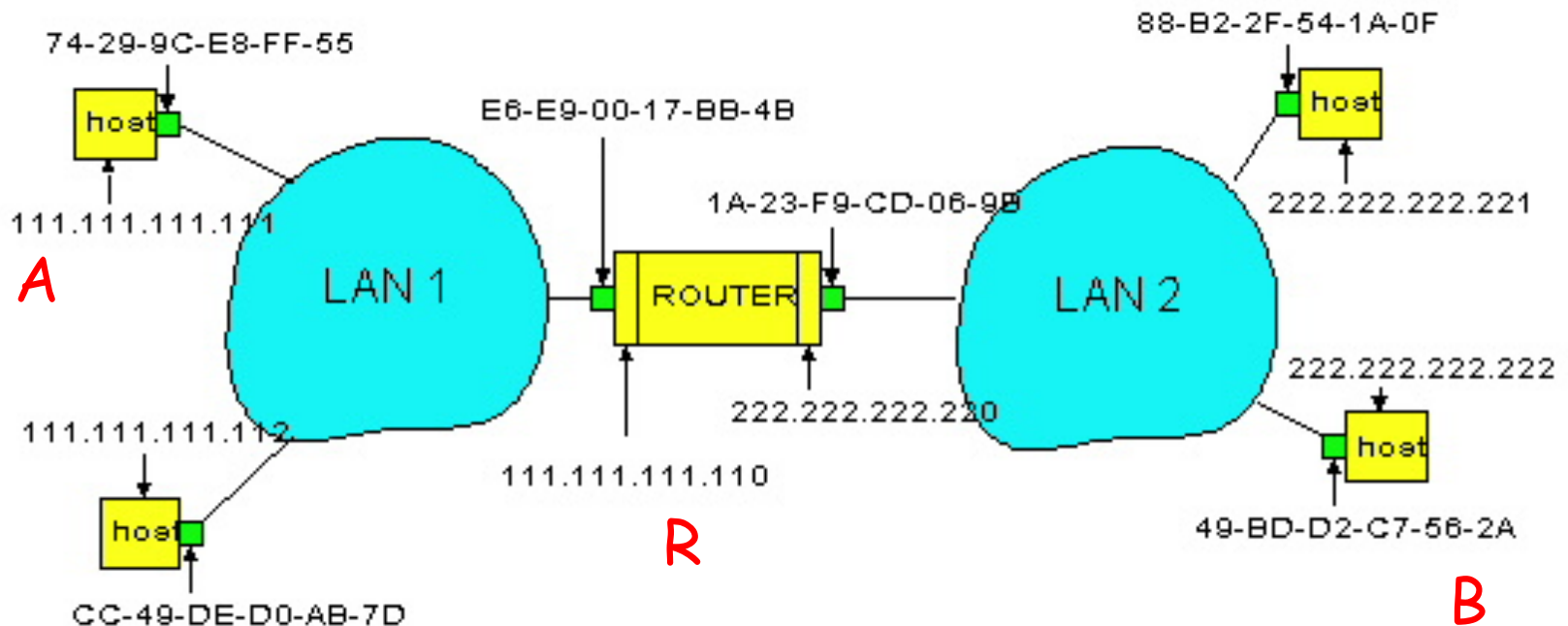
- Router *R*'s adapter receives the packet
 - ◆ *R* extracts the IP packet destined to 222.222.222.222
- Router *R* consults its forwarding table
 - ◆ Packet matches 222.222.222.0/24 via other interface



R Wants to Forward Packet



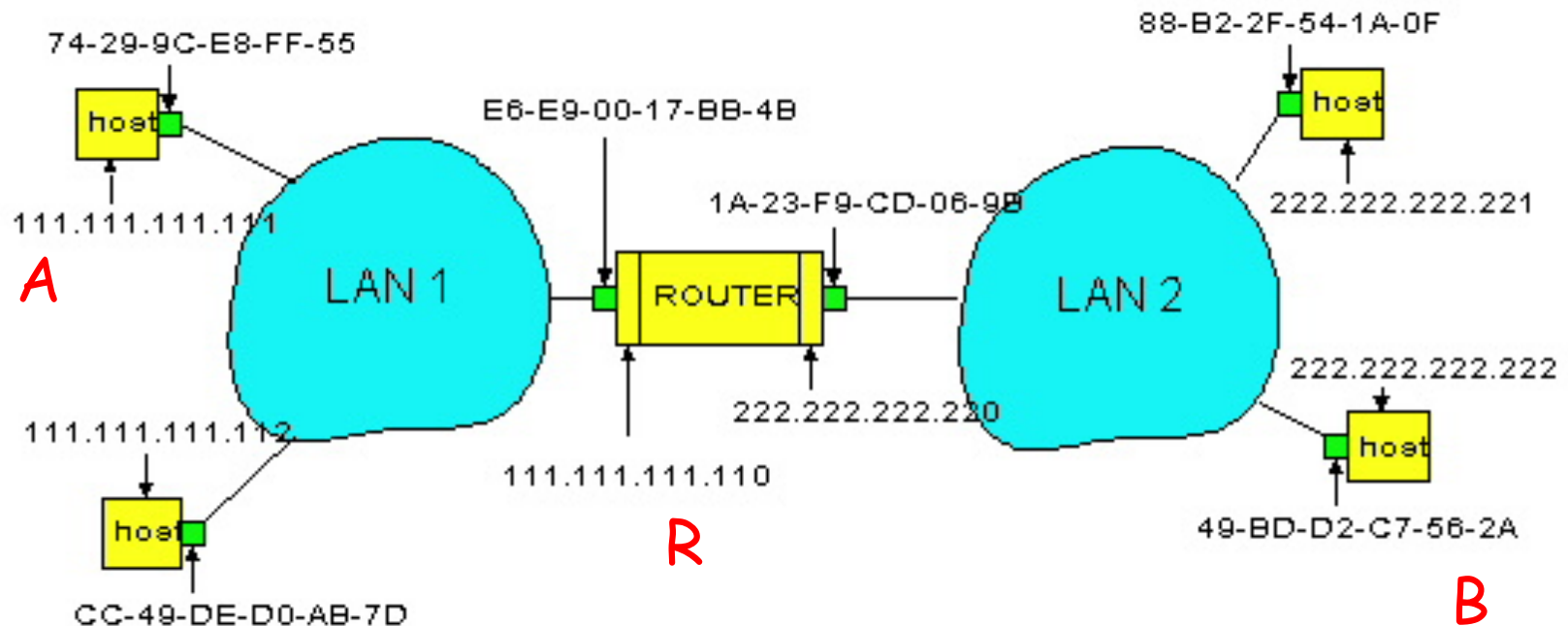
- IP packet
 - ◆ From A: 111.111.111.111
 - ◆ To B: 222.222.222.222
- Ethernet frame
 - ◆ From R: 1A-23-F9-CD-06-9B
 - ◆ To B: ???





R Sends Packet to *B*

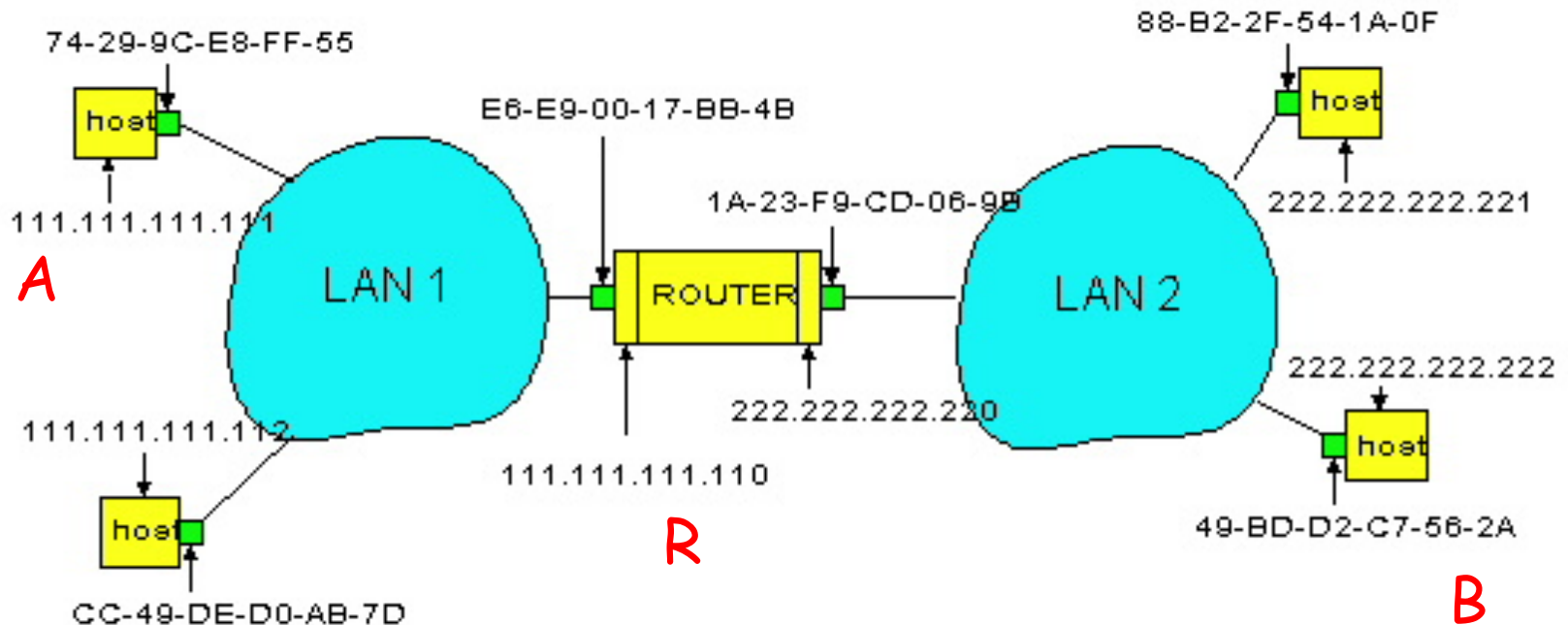
- Router *R*'s learns the MAC address of host *B*
 - ◆ ARP request: broadcast request for 222.222.222.222
 - ◆ ARP response: *B* responds with 49-BD-D2-C7-56-2A
- Router *R* encapsulates the packet and sends to *B*



R Wants to Forward Packet



- IP packet
 - ◆ From A: 111.111.111.111
 - ◆ To B: 222.222.222.222
- Ethernet frame
 - ◆ From R: 1A-23-F9-CD-06-9B
 - ◆ To B: 49-BD-D2-C7-56-2A





What's in a Router?

- Physical components
 - ◆ One or more **input interfaces** that receive packets
 - ◆ One or more **output interfaces** that transmit packets
 - ◆ A chassis (box + power) to hold it all
- Functions
 - ◆ **Forward** packets
 - ◆ **Drop** packets (congestion, security, QoS)
 - ◆ **Delay** packets (QoS)
 - ◆ **Transform** packets? (Encapsulation, Tunneling)

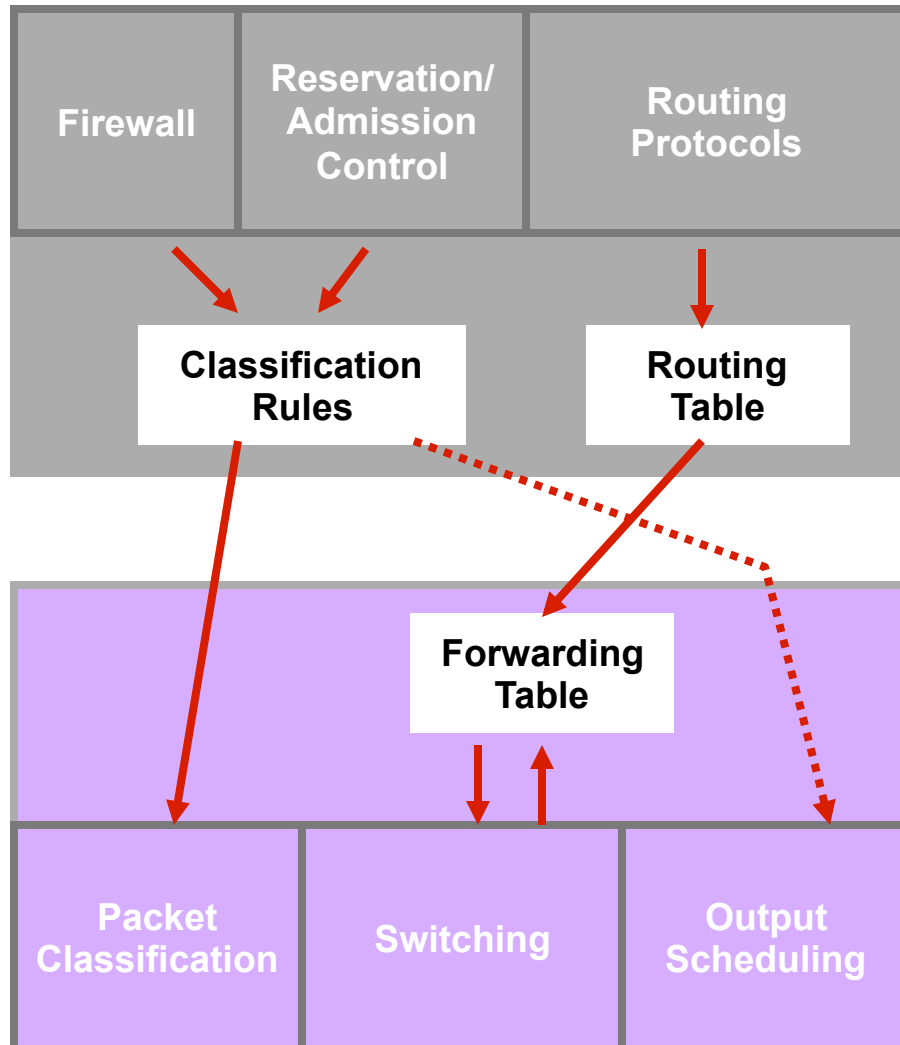


Router Functions

1. Receive incoming packet from link input interface
2. Lookup packet destination in forwarding table
(destination, output port(s))
3. Validate checksum, decrement ttl, update checksum
4. Buffer packet in input queue
5. Send packet to output interface (interfaces?)
6. Buffer packet in output queue
7. Send packet to output interface link



Functional architecture



Control Plane

- Complex
- Per-control action
- May be slow

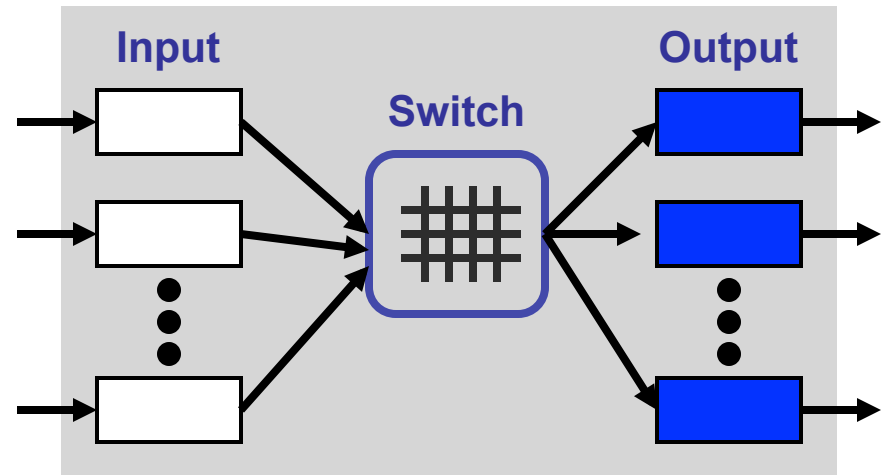
Data plane

- Simple
- Per-packet
- Must be fast



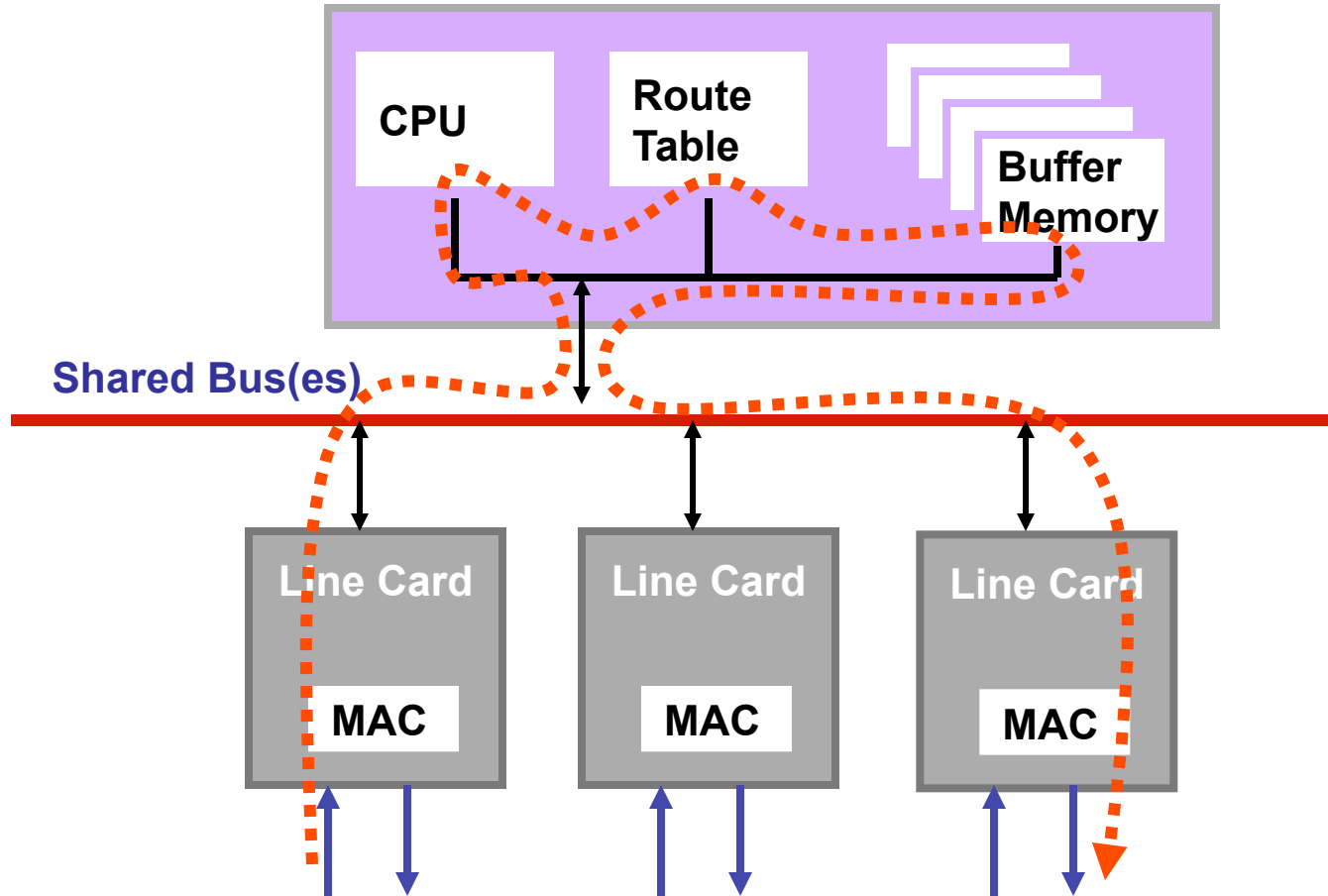
Interconnect architecture

- Input & output connected via switch fabric
- Kinds of switch fabric
 - ◆ Shared Memory
 - ◆ Bus
 - ◆ Crossbar
- How to deal with transient contention?
 - ◆ Input queuing
 - ◆ Output queuing





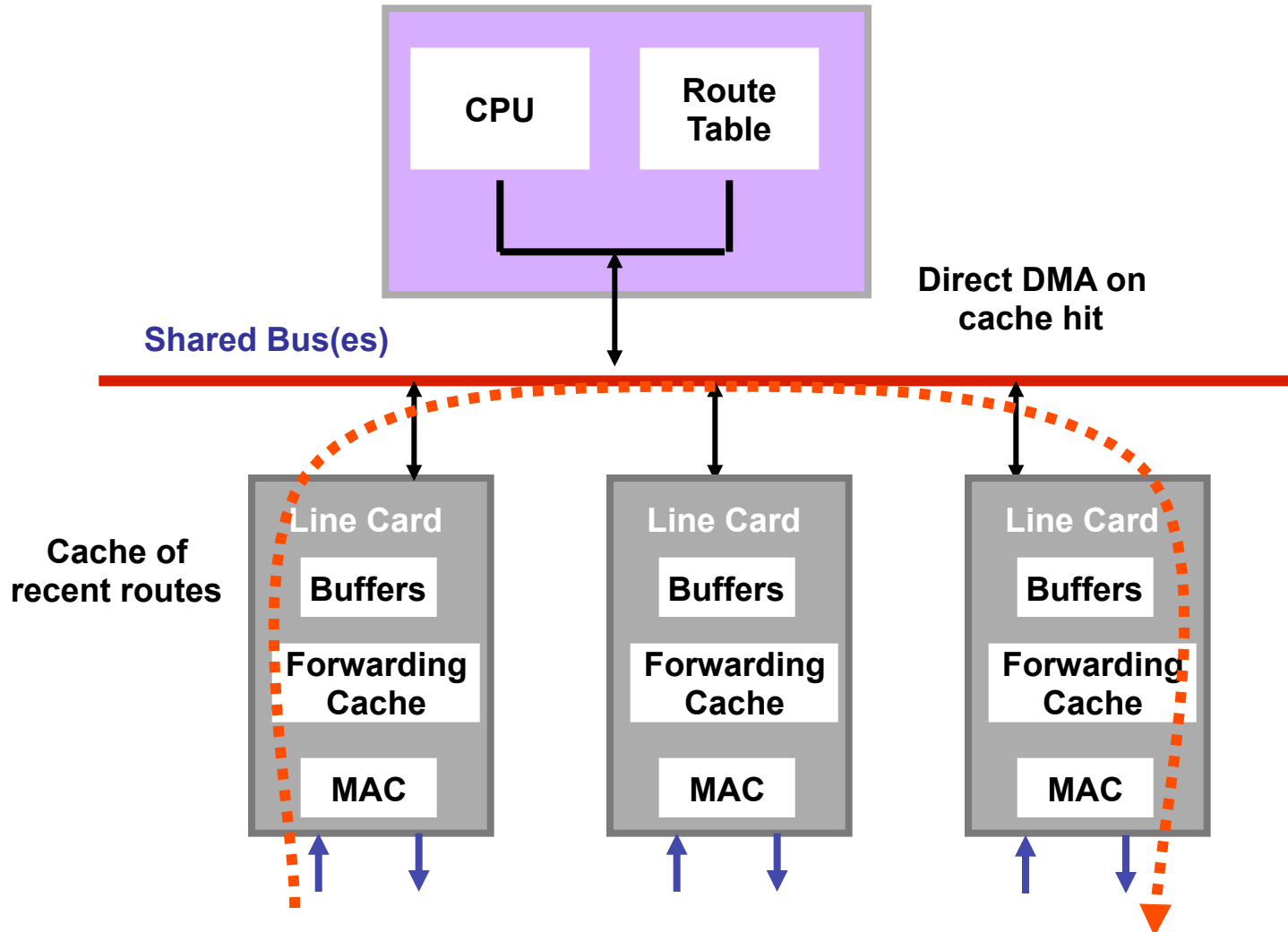
First Generation Routers



Single CPU and shared memory;

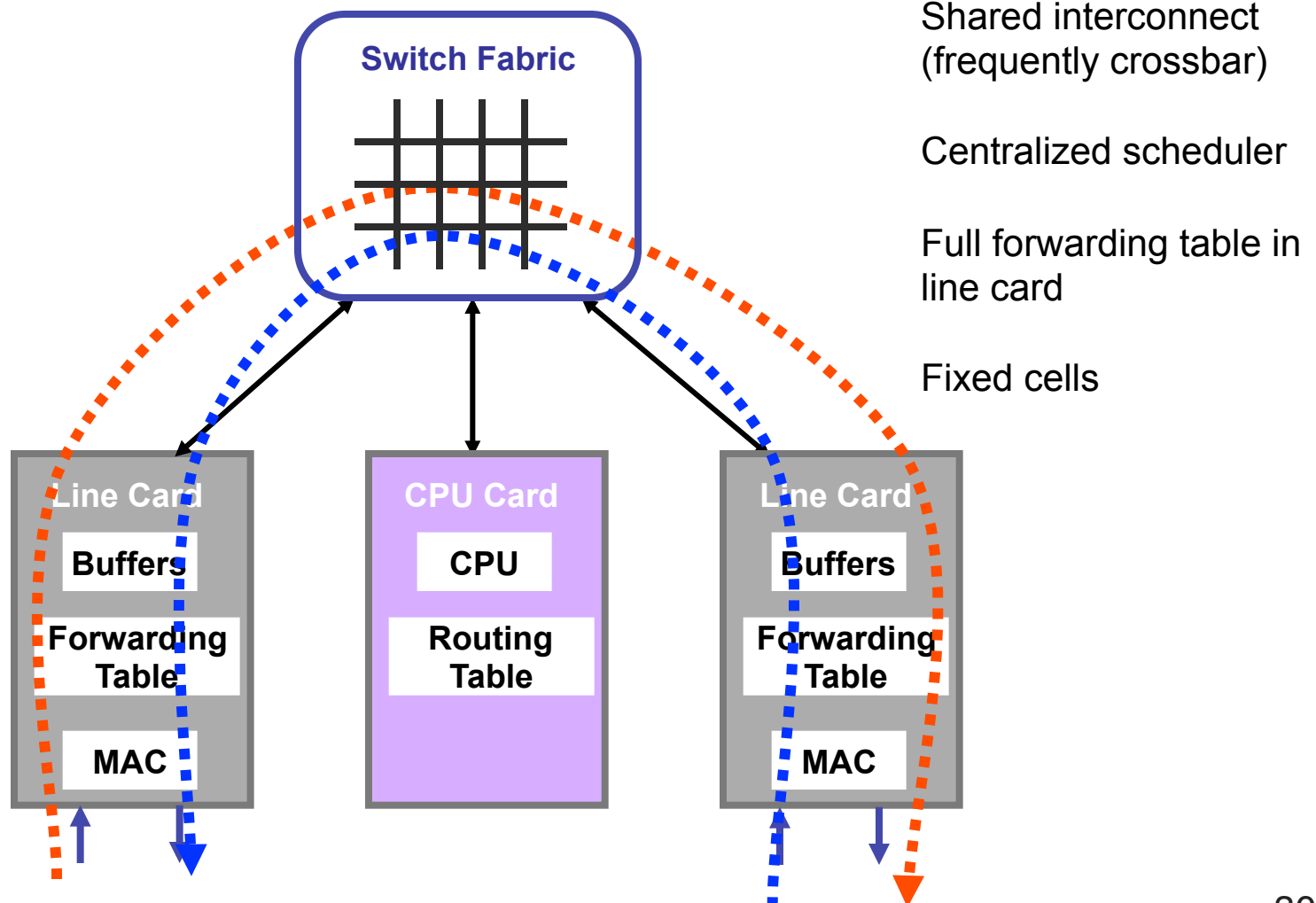
All classification by main CPU

Second Generation Routers





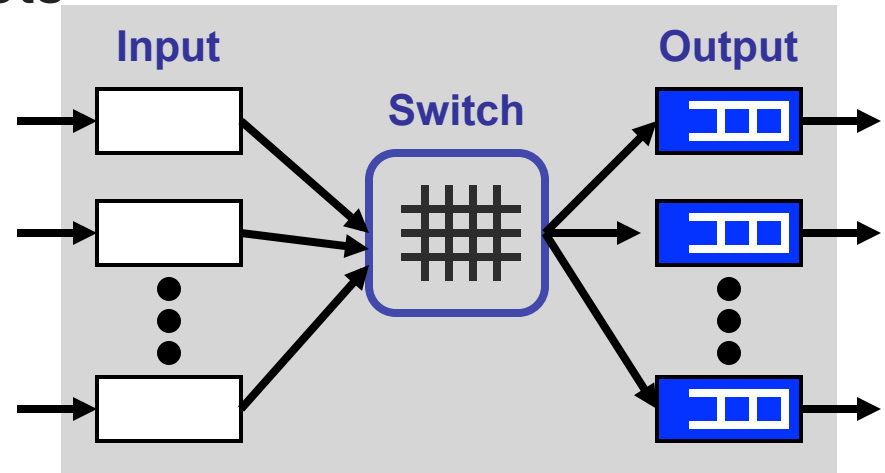
Third Generation Routers





Output queuing

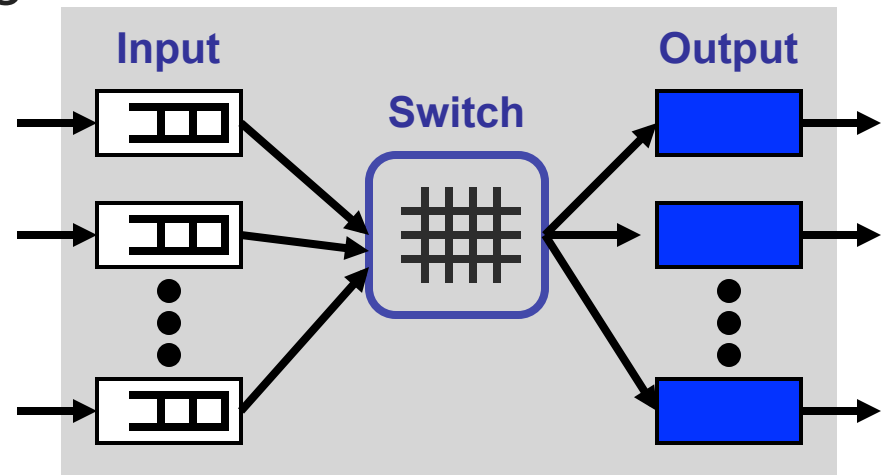
- Output interfaces buffer packets
- Pro
 - ◆ Simple algorithms
 - ◆ Single congestion point
- Con
 - ◆ N inputs may send to the same output
 - ◆ Requires **speedup** of N
 - » Output ports must be N times faster than input ports





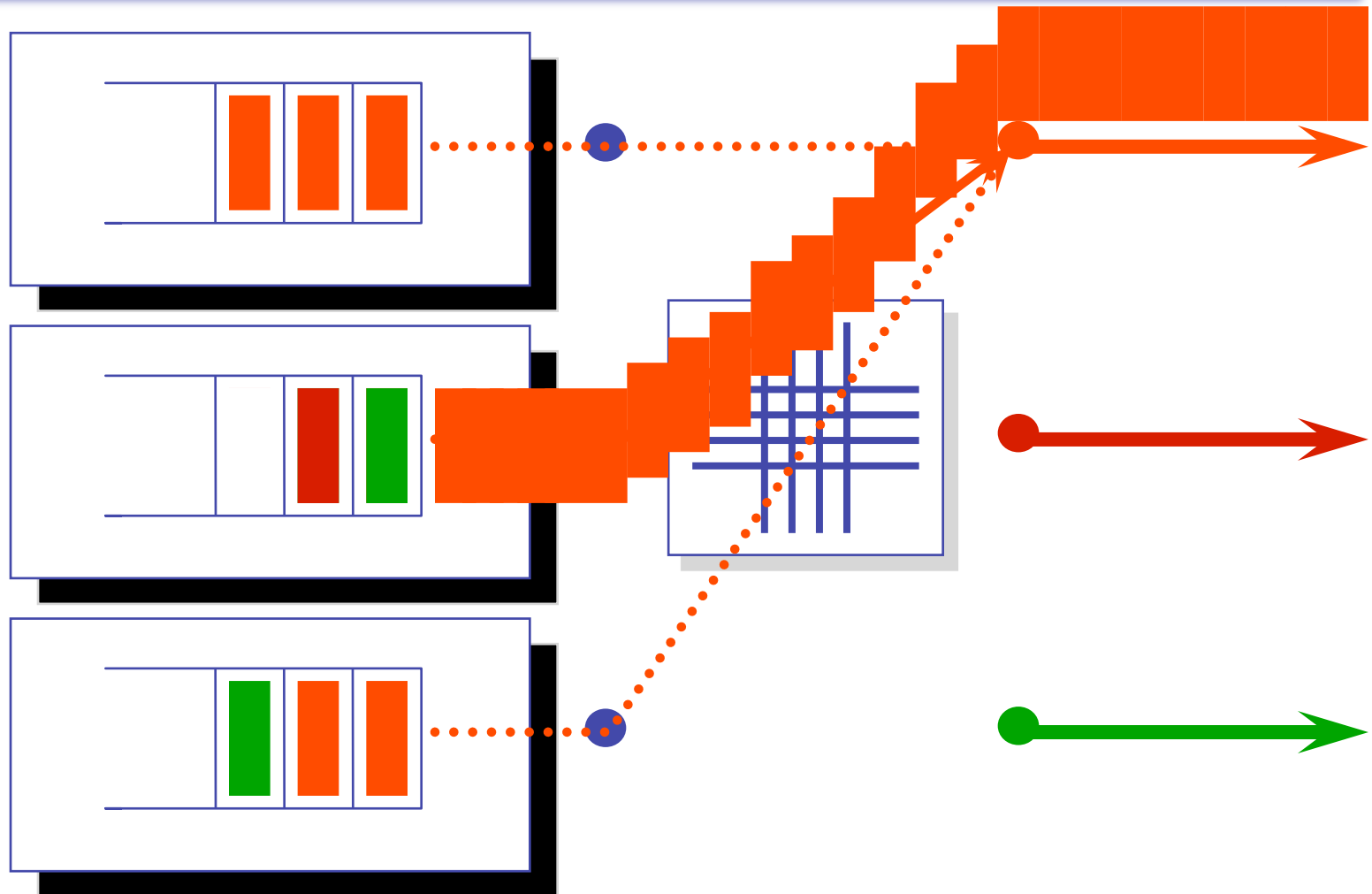
Input queuing

- Input interfaces buffer packets
- Pro
 - ◆ Single congestion point
 - ◆ Simple to design algorithms
- Con
 - ◆ Must implement flow control
 - ◆ Low utilization due to Head-of-Line (HoL) Blocking





Head-of-Line Blocking



IQ + Virtual Output Queuing



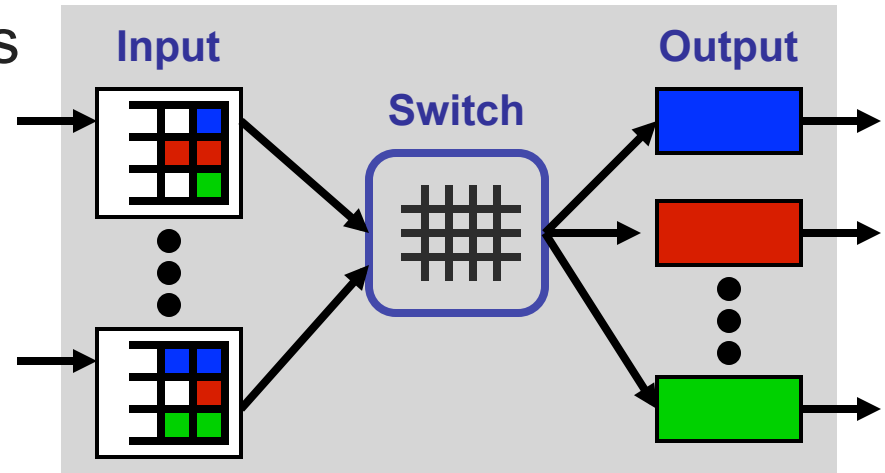
- Input interfaces buffer packets in per-output virtual queues

- Pro

- ◆ Solves blocking problem

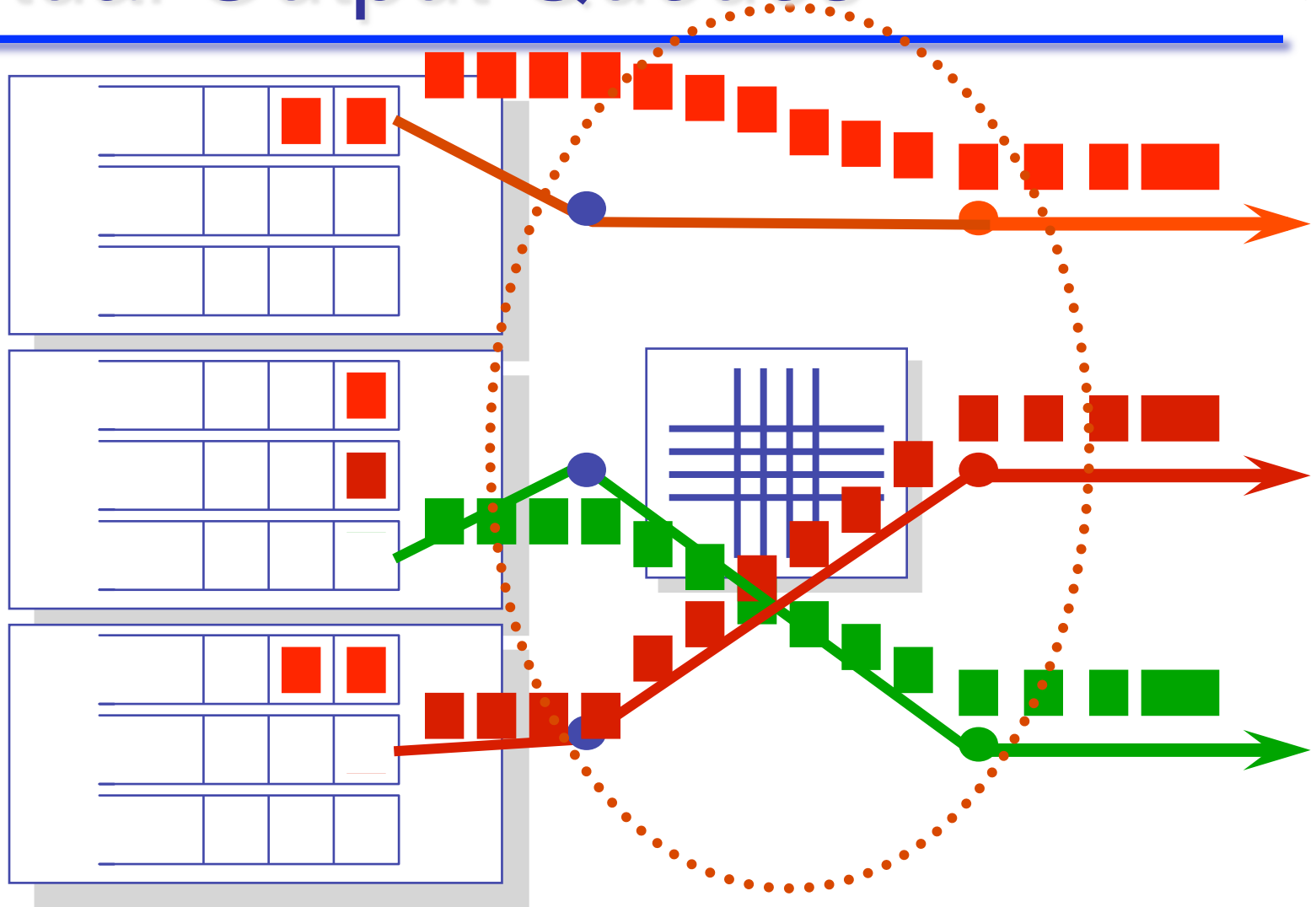
- Con

- ◆ More resources per port
 - ◆ Complex arbiter at switch
 - ◆ Still limited by input/output contention (scheduler)





Virtual Output Queues





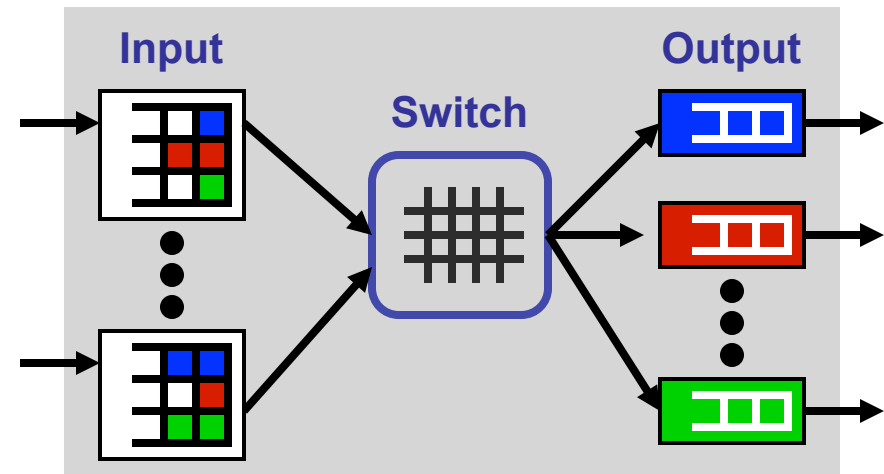
Switch scheduling

- Problem
 - ◆ Match inputs and outputs
 - ◆ Resolve contentions, no packet drops
 - ◆ Maximize throughput
 - ◆ Do it in constant time...
- If traffic is uniformly distributed its easy
 - ◆ Lots of algorithms (approximate matching)
- Seminal result (Dai et al, 2000)
 - ◆ Maximal size matching + *speedup* of two guarantees
100% utilization for most traffic assumptions



Typical high-performance router

- IQ + VoQ + OQ
 - ◆ Speedup of 2
 - ◆ Central scheduler
 - ◆ Fixed-sized internal cells
- Pro
 - ◆ Can achieve utilization of 1
 - ◆ Can scale to $> \text{Tb/s}$
- Con
 - ◆ Multiple congestion points
 - ◆ Complexity





For Next Time

- Read P&D 4.2